Students Mental Health Analysis

github: https://github.com/Lyn9527/DATA1030_Project.git

Lin Zhou
DSI
Brown University
Providence, RI
lin_zhou@brown.edu

1. Introduction

1.1 Purpose

Mental health plays a key role in students' overall well-being and academic success. However, mental health issues, especially depression, are often overlooked or untreated due to stigma, lack of awareness, or insufficient resources.[1] As academic pressure and external stressors increase, predicting and addressing students' mental health challenges becomes increasingly important.[2]

This project aims to predict students' mental health, specifically their depression scores, using a comprehensive dataset of academic, behavioral, and lifestyle factors. By employing advanced data pre-processing techniques and machine learning models, the goal is to identify key predictors of depression and provide insights that can aid in early detection and intervention. Ultimately, this research aims to contribute to the development of data-driven strategies to improve mental health support systems within educational institutions.

1.2 Dataset

The Students Mental Health Assessments dataset, sourced from Kaggle, represents mental health evaluations of students. This dataset seeks to provide valuable insights into the mental health of students by capturing a number of factors that may impact their mental health.

The dataset comprises a rich collection of records, carefully selected from various anonymous sources to ensure privacy and confidentiality. It's essential to acknowledge that no dataset is ever 100% accurate, as it can be affected by numerous sources of error and uncertainty.[3]

The dataset consists of 7,022 entries and 20 features covering a variety of demographic, academic, behavioral, and lifestyle factors.

- Demographic: Includes features such as Age, Gender, and Residence_Type
- Academic: Includes features like Course, CGPA, Semester_Credit_Load
- Personal and Behavioral: Includes features such as Relationshop_Status,Substance_Use,Counseling_Service_Use, Family_History,Chronic_Illness,Financial_Stress,and Extracurricular_Involvement.
- Mental Health and Lifestyle: Includes features like Depression_Score, Anxiety_Score, Stress_Level, Sleeping_Quality, Physical_Activity, Diety_Quality, and Social_Support.

The target variable for this project is Depression_Score, which is an ordinal variable ranked on a scale of 0 to 5, representing different degrees of depression severity. While Depression_Score can be modeled as a regression or classification problem due to its ordinal nature, this project focuses on regression approaches.

2. Exploratory Data Analysis

2.1 Missing Values

The first step in my EDA is to identify and quantify missing values within the dataset. After summarizing the dataset, I found that only two columns contain missing values:

1. CGPA (Numerical): Approximately 10.1538% missing.
2. Substance_Use (Categorical): Approximately 10.1832% missing.

Overall, the dataset has about 10.3389% missing values.

2.2 Distribution of Target Variable

To better understand the nature of the target variable, I explored its distribution. I plotted a distribution plot with a kernel density estimate overlay to visualize the frequency of different depression scores.

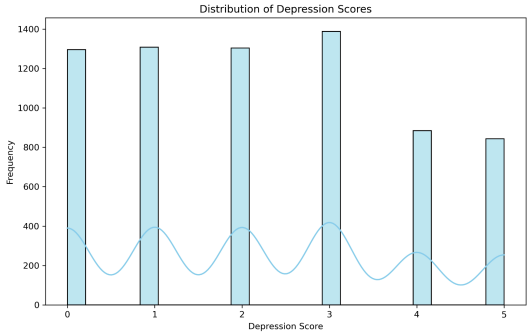

Figure 1: Distribution of Depression Scores

From the distribution plot, we can see that the depression scores are almost evenly distributed across the dataset, with no significant imbalance. However, the counts for depression scores of 4 and 5 are noticeably lower compared to the other scores (0, 1, 2, and 3). The near-even distribution is

[1] https://www.apa.org/monitor/2022/10/mental-health-campus-care
[2] https://www.nea.org/nea-today/all-news-articles/mental-health-crisis-college-campuses
[3] https://www.kaggle.com/datasets/sonia22222/students-mental-health-assessments/data

beneficial for training models because it ensures that most levels of depression are well represented.

2.3 Correlation Heatmap Between Target Variable and Numerical Features

To explore the relationship between the target variable Depression_Score and the numerical features in the dataset, I plotted a heatmap of their correlation.

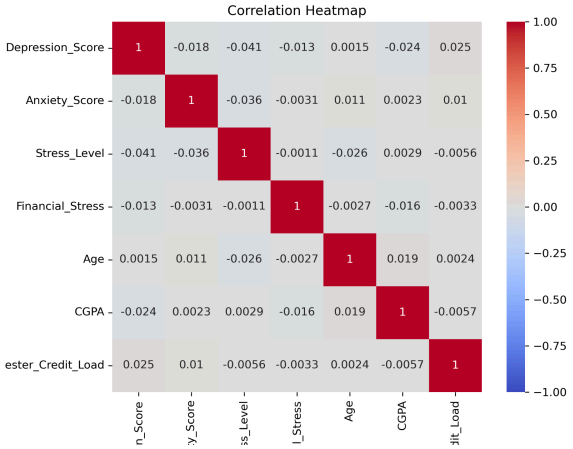

Figure 2: Heatmap between Target variables and Numerical Features

From the heatmap, we could find that almost all numerical features have weak correlation with Depression_Score, indicating that none of the numerical features strongly influence the depression scores linearly.

2.4 Heatmap Between Target Variable and Course

To explore the distribution of Depression_score based on the course, I plotted a heatmap of the standardized distribution of depression scores by course.

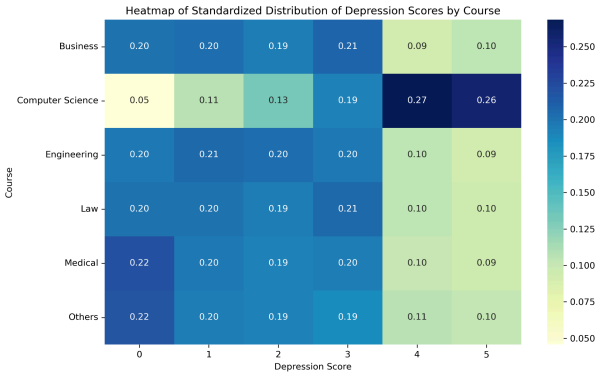

Figure 3: Heatmap of Standardized Distribution of Depression Scores by Course

From the heatmap, we could see that the students in Computer Science major tend to have higher depression score distribution, and the distribution of depression scores among different course is different, thus we could see that the dataset might be a course-specific dataset that the data survey might comes from different groups of students from different courses. This suggests that we might need to apply GroupShuffleSplit and GroupKFold to avoid data leakage.

2.4 Line plot of Depression Score Over CGPA

To analyze the relationship between students' academic performance and their average depression scores, I created a line graph showing the trend of average depression scores across different CGPA values.

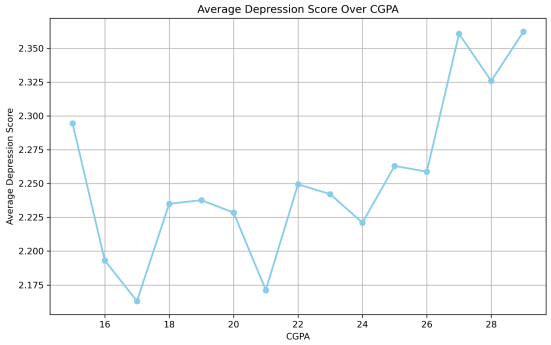

Figure 4: Line plot of Average Depression Score Over CGPA

From the line plot, we can see that students with higher CGPA would tend to have higher depression scores on average.

2.5 Overall distribution plots

To gain a comprehensive understanding of my dataset, I visualize the distribution of each feature using a combination of bar plots and KDE plots.

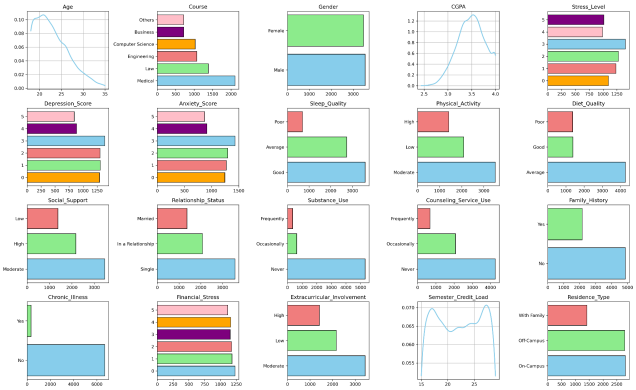

Figure 5: Overall distribution of each feature

3. Methods

3.1 Splitting Strategy

As I mentioned in the EDA, the dataset might be a course-specific dataset, thus there might be potential groups in my dataset. For all models, I performed a GroupShuffleSplit to split the dataset into (validation + train) : test = 8:2. Then I applied GroupKFold to split into train and validation sets with n_splits = 3. In this way, I got one true test set and three folds of train and validation sets.

3.2 Preprocessing

I applied Minmax scalar on numerical features such as CGPA, Age, and Semester_Credit_Load since they tend to have a lower and upper bound.
For categorical features, I applied OneHotEncoder. Notice that one of the features that contain missing values is Substance_Use, which is an ordinal variable. Thus, for ordinal features, I applied OrdinalEncoder with handle_unknown = 'use_encoded_value' and set unknown_value = 1. In this way, all the missing values in Substance_Use are treated as -1. For missing values in CGPA, I simply drop rows containing missing values for models except XGBoost since XGBoost can handle missing values on its own.

3.3 ML pipeline

There are overall 6 ML algorithms used in this project (3 linear and 3 non-linear): Linear Regression with no regularization (serve as baseline model), Linear Regression with Elastic Net, SVM with kernel function set to linear, Random Forest, KNN, and XGBoost. For each algorithm, several hyperparameters are tuned: (Table 1)

ML Algorithm	Hyperparameters	Best Parameters
Linear Regression(no regularization)	None	None
Linear Regression (Elastic Net)	alpha: [0.01, 0.1, 0.5, 1, 2, 5, 10], l1_ratio: [0.1, 0.5, 0.7, 0.9, 1]	alpha: 100 l1_ratio: 1
SVM(kernel = 'linear')	c: [1e-2, 1e-1, 1e0, 1e1, 1e2], gamma: [1e-3, 1e-1, 1e1, 1e3, 1e5]	c: 0.01 gamma: 0.001
Random Forest	max_depth: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], max_features: ['sqrt', 'log2']	max_depth: 1 max_features: log2
KNN	n_neighbors: [9, 11, 25, 100, 350, 400, 450], weights': ['uniform', 'distance'], p: [1, 2]	n_neighbors: 450 p: 1 weights: distance
XGBoost (learning rate = 0.03)	reg_alpha: [0e0, 1e-2, 1e-1, 1e0, 1e1, 1e2], reg_lambda: [0e0, 1e-2, 1e-1, 1e0, 1e1, 1e2], max_depth: [1,3,10,30,100]	max_depth: 1 reg_alpha: 100 reg_lambda: 100

Table 1: Tuned hyperparameter and best performed value for each model

The evaluation metric for each model is RMSE. Although the target variable is ordinal, we cannot use Accuracy, f1 scores as our evaluation metric. This is because, for example, if the true value is 2 the penalty for the model to predict 3 should be different from the one to predict 5. The latter one should have a greater penalty. Thus, we should use RMSE for the ordinal nature of my target variable. It would give us a better understanding as well as better performance for our models.

In the pipeline, each model is assigned five different random states when splitting. At the end of each loop, we save the best_model, best_testscore, results, y_test_pred, y_test, x_test, and the grid. I used the best_test_score to compare the performance among different models, as the random states result in different test scores for the same model. The others are used for future feature importance analysis.

3.4 Uncertainties of each model

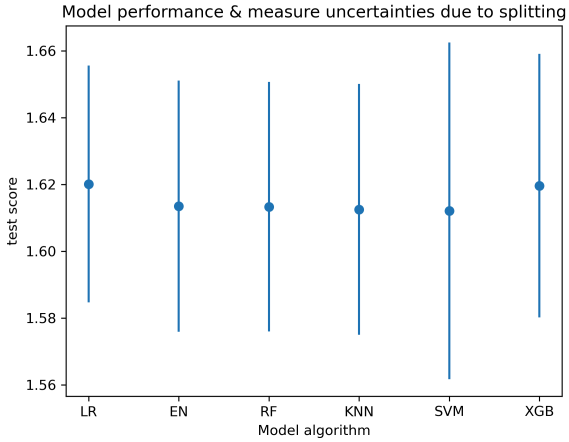

Figure 6: Measure of uncertainties of each model

Here is the plot of uncertainties of each model due to splitting under 5 different random states. We can see here that the SVM has the greatest uncertainty, while others tend to have similar uncertainty.

4. Results

4.1 performance of each model

ML Algorithm	Mean test scores	Std test scores	Std over baseline
Linear Regression(no regularization)	1.620149	0.035456	-0.081025
Linear Regression (Elastic Net)	1.613475	0.037585	0.101134
SVM(kernel = 'linear')	1.612091	0.050386	0.102903
Random Forest	1.613336	0.037328	0.105533
KNN	1.612559	0.037586	0.125491
XGBoost (learning rate = 0.03)	1.619649	0.039424	-0.060186
Baseline	1.617276	0.000000	0.000000

Table 2: mean, std, and how many std away from the baseline for each models

I setted mean of the depression score as my baseline prediction and calculated the baseline (1.617276). We can see from table 2 here, most models have slightly improved from baseline, however XGBoost and linear regression with no regularization are worse than the baseline. Among all the models, SVM has the lowest mean test score, indicating it is the best model for my project. And I will use this model for future global and local feature importance analysis.

4.2 Global Feature Importance

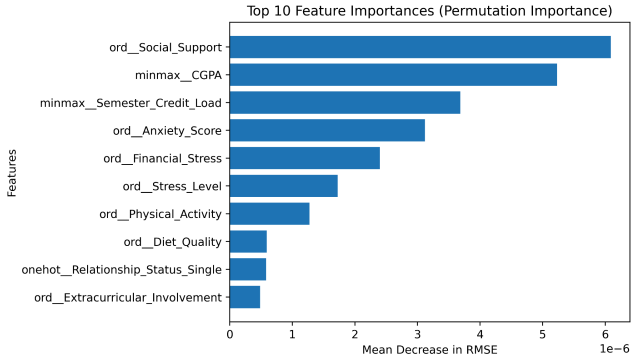

Figure 7: Top 10 Features By Permutation Importance

First, I applied the permutation importance to check the top 10 important features. It shows that social_support the most important feature. It indicates that students who need social support would tend to have higher depression scores.

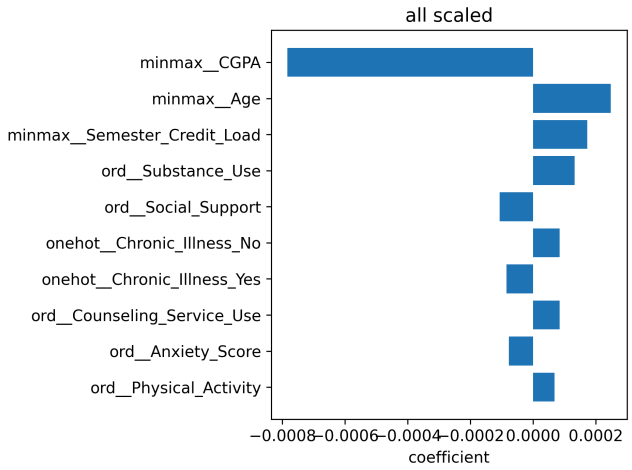

Figure 8: Top features By Linear Coefficient

Since the best model is SVM with kernel function set to linear, we could check the linear coefficient to find out which is the most important feature. We can see here the CGPA is the most important feature.

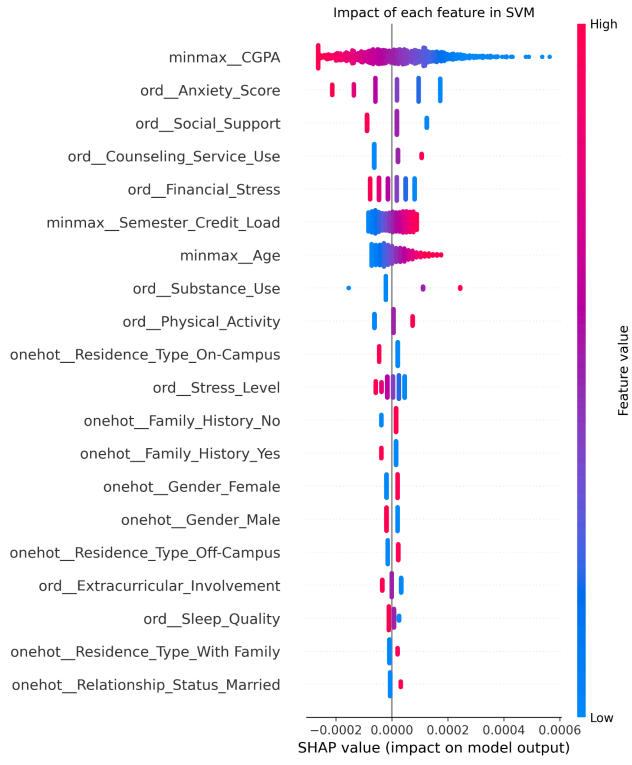

Figure 9: SHAP

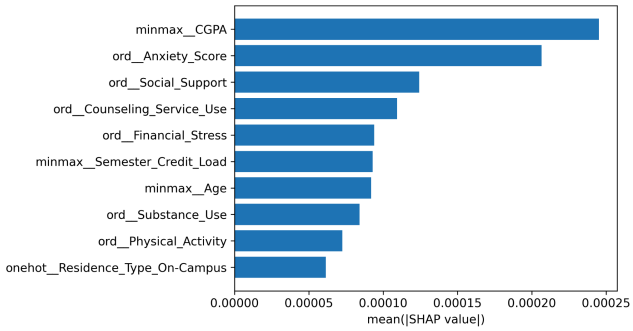

Figure 10: Mean Absolute of SHAP Value

We can see from figure 9 and 10, I find the most important features through SHAP values. The result is similar to the linear coefficient ones in that CGPA is the most important feature. It indicates that the students with higher CGPA would tend to have higher depression scores.

Among all these three techniques computing the global feature importance, CGPA is always in top 3, showing it contributes the most to the prediction, where features like residence_type, relationship_status, and genders have least feature importances as they appear at the bottom of the importance figures.

4.3 Local Feature Importance

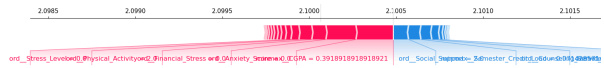

Figure 11: Local Feature Importance for index = 20

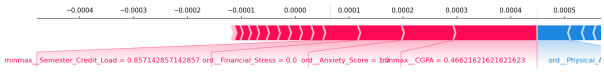

Figure 12: Local Feature Importance for index = 35

Here I randomly picked two indexes for analysing the local feature importance using SHAP. We can see here features like CGPA, Anxiety_score, Financial_Stress, and Semester_Credit_Load are pushing the prediction higher. It means that these academic and personal factors are major contributors to elevated student's depression levels. Besides, we can see that features like Physical_Activity, Residence_Type, and etc. are trying to pull the prediction lower, indicating that these features are positively affecting students' mental health.

5. Outlook

When dealing with missing values, except XGBoost, I simply dropped the rows containing missing values. For further improvement, I could use the reduced-feature model to handle the missing values in CGPA. In this way, no rows would be dropped and all the information would be kept.

Furthermore, since my target variable is ordinal, I could try the classification approach to predict the depression score and use other evaluation metrics such as MAE, a combination of f1 score and Accuracy, and weighted Kappa to see whether there is an improvement in prediction.

Moreover, since most models in my project are designed for basic regression or classification problems, which might not perform well on ordinal target variables. I could try some models that suit for ordinal variables such as CATBoost[4]. Or manually implement some ML algorithms that suit the ordinal target variable.

6. Reference

[1] American Psychological Association. (2022, October). *The state of mental health on college campuses*. Monitor on Psychology. Retrieved from https://www.apa.org/monitor/2022/10/mental-health-campus-care

[2] National Education Association. (n.d.). *Addressing the mental health crisis on college campuses*. NEA Today. Retrieved from https://www.nea.org/nea-today/all-news-articles/mental-health-crisis-college-campuses

[3] Sonia22222. (n.d.). *Students mental health assessments [Data set]*. Kaggle. Retrieved from https://www.kaggle.com/datasets/sonia22222/students-mental-health-assessments/data

[4] CatBoost. (n.d.). *Ranking loss functions*. Retrieved from https://catboost.ai/docs/en/concepts/loss-functions-ranking

[4] https://catboost.ai/docs/en/concepts/loss-functions-ranking